A Wavelet Neural Network based approach in Cancer Diagnosis and Biopsy Classification*

[1]Akarsh Sinha, [1]M. Pavithra and [1]B. Jaganatha Pandian

[1]School of Electrical Engineering, VIT University, Vellore, India;
Email: akarsh.sinha2010@vit.ac.in

ABSTRACT

Neural network has been playing a major role in medical electronics in classifying normal and abnormal cases by extracting features from various measurements. It is necessary to design an efficient network to relate the features extracted from biopsy analysis into cancerous and non-cancerous classes. To improve the efficiency of this diagnosis wavelet based neural network is proposed in this paper. Mother wavelets like Morlet, Shannon and Polywog were used to train the Wavelet Neural Network (WNN) in this paper. The WNN learns the nonlinear functions faster. Also the classification using WNN reduces false positives and false negatives considerably.
Keywords : Biopsy, Cancer Diagnosis, Morlet, Polywog, Shannon, Wavelet Neural Network

1 INTRODUCTION

EARLYdetection of cancer plays a major role in increasing the survival rate of patients. Artificial Intelligent techniques are widely used in medical field to handle such classification and estimation problems.

The purpose of this paper is to classify a cancer dataset into benign and malignant using wavelet neural network and compare network's performance with a feed forward network trained using Levenberg - Marquardt back propagation algorithm. The cancer dataset consists of features extracted from biopsy analysis.

Wavelet neural network combines the ability of wavelet decomposition and the performance of neural network. According to [1] and [3] the quality of approximation can be the same as that of a feed forward network with reduced network size. In wavelet neural networks we use the wavelet function as the activation function in the hidden nodes.

The comparative study present in the paper is done upon identical network structures with different hidden layer activation function. The number of neurons in the input layer is 9, in hidden layer is 17 and in the output layer is 2. The activation functions used for wavelet neural network are Morlet, Shannon and Polywog. The simple feed forward network was trained with tan-sigmoid activation function.

2 MATERIALS AND METHOD

2.1 Cancer dataset

The breast cancer dataset provided along with The Neural Network Toolbox, MATLAB® package has been used in this paper. This data is available from the UCI Machine Learning Repository [2]. The dataset consists of nine attributes of 699 biopsy cases and their corresponding classification of either Benign or Malignant. There are 458 Benign and 241 malignant cases.

The nine attributes given are, Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal Adhesion, Single epithelial cell size, Bare nuclei, Bland chromatin, Normal nucleoli and Mitoses.

Clump thickness*:* Cancerous cells are often grouped in multilayer. This feature presents the thickness of such multi-layer.
Uniformity of cell size/shape*:* Cancer cells tend to vary in size and shape for benign and malignant cases.
Marginal adhesion: This is the measure of adhesive property between cells.
Single epithelial cell siz*e:* This is the measure of epithelial cell size. For a malignant case this size will be high.

ISSN 2278-7763

Bare nuclei*:* This is the measure of nuclei that is not surrounded by cytoplasm.
Bland Chromatin: This describes the uniformity of the "texture" of the nucleus. For malignant cases this will be poor.
Normal nucleoli*:* This is the measure of nucleolus size. For a normal case it's small and will be prominent for cancerous cells
Mitoses: This feature indicates the mitotic activity. For a cancerous lump this will be more.

2.2 WAVELET NETWORK

The wavelet neural networks are highly suitable for approximating arbitrary non-linear functions. The structure of the three layer wavelet neural networks used in this paper is given in Fig. 1.

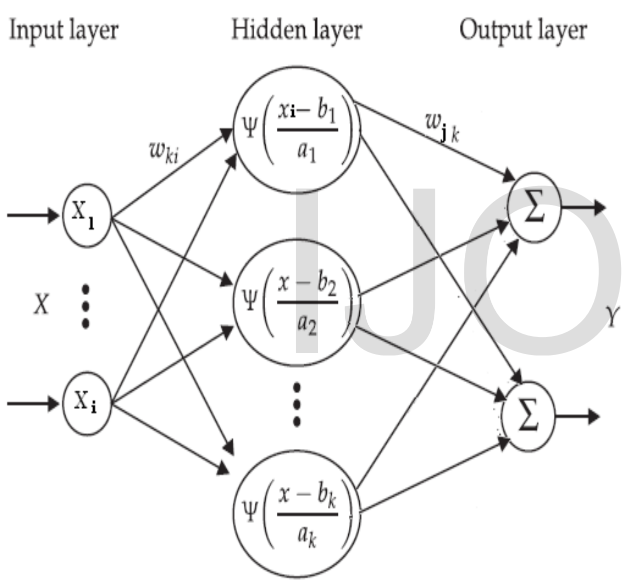


Fig.1 Wavelet neural network structure

In Fig. 1.

$\mathbf{w_{ki}}$ –	Weight between input and hidden layer
$\mathbf{w_{jk}}$ –	Weight between hidden and output layer
X, Y –	Input and Output vectors
$\Psi(x)$ –	Wavelet function
$\mathbf{a_k}$ –	Dilation parameter of k^{th} hidden neuron
$\mathbf{b_k}$ –	Translation parameter of k^{th} hidden neuron

The steps involved in training the WNN is given below.

Step 0: Initialize the WNN parameters to random value.

Step 1: Present the input pattern and target pattern to WNN.

Step 2: Compute the WNN output

$$\mathbf{Y}_j = \sum_k \mathbf{w_{jk}}\, \psi\left(\frac{\sum_i \mathbf{w_{ki}} \mathbf{x_i} - \mathbf{b_k}}{\mathbf{a_k}}\right) \qquad (1)$$

Step 3: Compare the output with the target and compute the sum of square of error.

Step 4: If the target value of sum of square of error is not reached at the 'n'th epoch, find the difference between the target and the output obtained (E) and compute Δw_{jk}, Δw_{ki}, Δb and Δa using the learning rate (η) and the momentum (α).

$$\Delta \mathbf{w_{ki}}(\mathbf{n}) = -\eta \frac{\partial \mathbf{E}}{\partial \mathbf{w_{ki}}} + \alpha\, \Delta \mathbf{w_{ki}}(\mathbf{n-1}) \qquad (2)$$

$$\Delta \mathbf{w_{jk}}(\mathbf{n}) = -\eta \frac{\partial \mathbf{E}}{\partial \mathbf{w_{jk}}} + \alpha\, \Delta \mathbf{w_{jk}}(\mathbf{n-1}) \qquad (3)$$

$$\Delta \mathbf{a_k}(\mathbf{n}) = -\eta \frac{\partial E}{\partial a_k} + \alpha\, \Delta \mathbf{a_k}(\mathbf{n-1}) \qquad (4)$$

$$\Delta \mathbf{b_k}(\mathbf{n}) = -\eta \frac{\partial \mathbf{E}}{\partial \mathbf{a_k}} + \alpha\, \Delta \mathbf{b_k}(\mathbf{n-1}) \qquad (5)$$

Step 5: Using this update the WNN parameters w_{jk}, w_{ki}, b and a for the next epoch

$$\mathbf{w_{ki}}(\mathbf{n+1}) = \mathbf{w_{ki}}(\mathbf{n}) + \Delta \mathbf{w_{ki}}(\mathbf{n}) \qquad (6)$$

$$\mathbf{w_{jk}}(\mathbf{n+1}) = \mathbf{w_{jk}}(\mathbf{n}) + \Delta \mathbf{w_{jk}}(\mathbf{n}) \qquad (7)$$

$$\mathbf{a_k}(\mathbf{n+1}) = \mathbf{a_k}(\mathbf{n}) + \Delta \mathbf{a_k}(\mathbf{n}) \qquad (8)$$

$$\mathbf{b_k}(\mathbf{n+1}) = \mathbf{b_k}(\mathbf{n}) + \Delta \mathbf{b_k}(\mathbf{n}) \qquad (9)$$

Step 6: Repeat steps 1-5 till convergence is achieved.

The wavelets functions employed in this work are Morlet, Shannon and Polywog. The wavelet name and the wavelet function used in equation (1) are given in Table-I.

Table 1
LIST OF WAVELET FUNCTIONS

Name	Function, $\psi(x)$
Morlet	$\cos(0.75x)+e^{-0.5x}$
Shannon	$\dfrac{\sin(1.75x)-\sin(0.75x)}{0.75x}$
POLYWOG1	$xe^{-0.5x^2}$

3 RESULTS

For validating the proposed classifier 699 biopsy cases from database [2] were considered. Out of the 699 patterns 659 patterns were used for training the networks and 40 of them used for testing the trained networks.

A feed forward neural network having a hidden layer of 17 neurons with tan sigmoid activation function was trained with 0.01 learning rate. The same training patters were used for training a wavelet neural network with 0.01 learning rate. The network has a hidden layer of 17 neurons with wavelet activation functions.

All networks have 2 linear neurons in the output layer to classify benign and malignant cases. The corresponding neuron output will be 1 and that of the other will be 0 for each case.

The convergences of sum of square of error with respect to epochs while training the networks are shown in Fig. 2.

The training graphs indicate the wavelet networks converge faster than an ordinary neural network. Among the wavelet networks, the network with Morlet function converges faster.

The obtained outputs from all networks for the 40 test patterns (20 – benign, 20 – malignant) are in Table –II and Table – III. While testing the Morlet WNN and Shannon WNN were giving results much closer to the expected values while Polywog WNN was performing almost similar to the simple ANN with tan sigmoid activation function.

4 CONCLUSION

From the results, obtained while training and testing a wavelet neural network and a feed forward neural network, we could see the wavelet neural network trains faster than a feed forward neural network. Also the closeness to the expected target is better in a wavelet neural network than a feed forward neural network. These results infer wavelet networks will perform efficiently in patterns classification and function approximation applications.

ACKNOWLEDGEMENTS

We thank the School of Electrical Engineering, VIT University, for providing us with the resources necessary to complete this project. We also thank our family for their invaluable support and inspiration.

REFERENCES

[1] Q. Zhang and A. Benveniste. 'Wavelet networks', IEEE Transactions on neural networks, volume 3, pages 889–898, 1992.

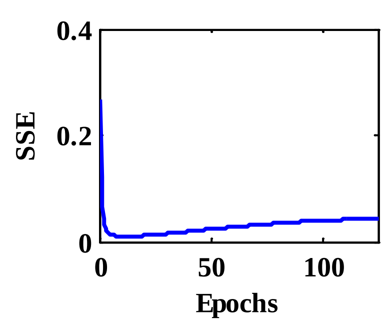
[2] Asuncion, A & Newman, D.J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[3] Licheng Jiao, 'Multiwavelet Neural Network And Its Approximation Properties', IEEE Transactions on neural networks, VOL. 12, NO. 5, 1060 – 1066 (2001).

[4] Howida Ali AbdElgader and Mohammed HassenHamza, 'Breast Cancer Diagnosis Using Artificial Intelligence Neural Networks', J.Sc. Tech, 159-163 (2011).

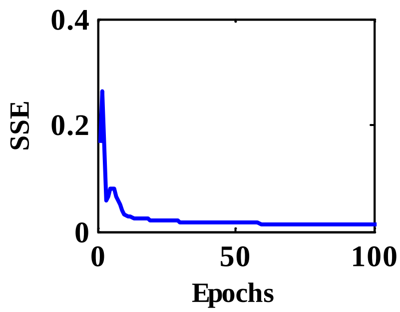
[5] Cheng-Jian Lin, 'Wavelet Neural Networks with a Hybrid Learning Approach', Journal Of Information Science And Engineering 22, 1367-1387 (2006).

[6] M. Unser, "Wavelets, Statistics, and Biomedical Applications," *Proceedings of the Eighth IEEE Signal Processing Workshop on Statistical Signal and Array Processing, pp. 244-249, June 1996.*

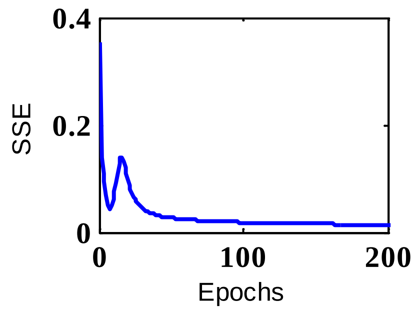


(a)

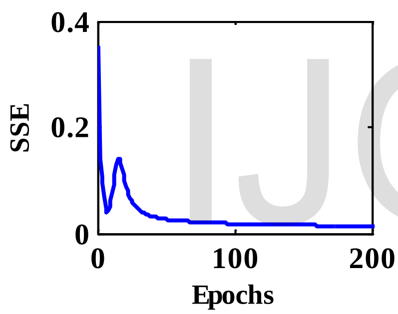
ISSN 2278-7763



(b)



(c)



(d)

Fig. 2. Plot of sum of square of error Vs. epoch number. (a) Feed Forward Network with Levenberg - Marquardt algorithm (b) Morlet WNN (c) Shannon WNN and (d) Polywog

Table 2
EXPECTED AND OBTAINED OUTPUTS FOR BENIGN CASES

Expected		ANN		Morlet WNN		Shannon WNN		Polywog WNN	
Node 1	Node 2	Node 1	Node 2	Node 1	Node 2	Node 1	Node 2	Node 1	Node 2
1	0	0.9600	0.0412	0.9995	-0.0009	0.9958	-0.0010	0.9410	0.0748
1	0	0.9276	0.0708	1.0010	-0.0018	1.0058	-0.0035	0.9398	0.0689
1	0	0.9278	0.0717	1.0011	0.0001	1.0013	-0.0028	0.9398	0.0746
1	0	0.9377	0.0632	1.0014	-0.0039	1.0010	0.0044	0.9398	0.0894
1	0	0.9368	0.0635	1.0018	-0.0017	0.9989	0.0038	0.9398	0.0746
1	0	0.9318	0.0689	0.9980	0.0011	0.9966	-0.0003	0.9407	0.0749
1	0	0.9443	0.0541	0.9977	0.0003	1.0003	0.0016	0.9398	0.0752
1	0	0.9397	0.0596	0.9977	0.0014	1.0006	0.0027	0.9380	0.0756
1	0	0.9397	0.0596	0.9977	0.0014	1.0006	0.0027	0.9417	0.0610
1	0	0.9397	0.0596	0.9977	0.0014	1.0006	0.0027	0.9390	0.0761
1	0	0.9364	0.0627	0.9898	0.0102	1.0026	-0.0006	0.9395	0.0752
1	0	0.9320	0.0676	1.0104	-0.0100	0.9997	0.0029	0.9390	0.0758
1	0	0.9325	0.0668	1.0094	-0.0100	1.0038	-0.0040	0.9388	0.0573
1	0	0.9190	0.0816	1.0095	-0.0076	0.9917	0.0080	0.9396	0.0437
1	0	0.9428	0.0600	0.9888	0.0093	1.0012	0.0020	0.9388	0.0763
1	0	0.9523	0.0470	0.9882	0.0112	1.0021	-0.0037	0.9378	0.0747
1	0	0.8152	0.1905	0.6720	0.3337	0.7400	0.2655	0.9316	0.0733
1	0	0.9277	0.0740	1.0939	-0.0945	1.0062	-0.0019	0.9377	0.0764
1	0	0.9345	0.0639	0.8994	0.1019	0.9880	0.0081	0.9390	0.0760
1	0	0.9321	0.0665	0.8940	0.1076	0.9824	0.0138	0.9388	0.0764

Table 3
EXPECTED AND OBTAINED OUTPUTS FOR MALIGNANT CASES

Expected		ANN		Morlet WNN		Shannon WNN		Polywog WNN	
Node 1	Node 2	Node 1	Node 2	Node 1	Node 2	Node 1	Node 2	Node 1	Node 2
0	1	0.0819	0.9180	0.0008	0.9996	-0.0898	1.0663	0.0748	0.9390
0	1	0.0767	0.9304	0.0015	1.0002	-0.0787	1.0589	0.0689	0.9398
0	1	0.0761	0.9277	0.0016	0.9971	-0.0901	1.0692	0.0746	0.9372
0	1	0.0783	0.9196	0.0025	0.9961	-0.0716	1.0756	0.0894	0.9305
0	1	0.0827	0.9138	-0.0019	0.9997	-0.0461	1.0697	0.0746	0.9349
0	1	0.0683	0.9286	-0.0028	1.0027	-0.0588	1.0602	0.0749	0.9390
0	1	0.0718	0.9263	0.0038	0.9979	-0.0730	1.0580	0.0752	0.9383
0	1	0.0574	0.9429	0.0067	0.9940	0.0491	0.9646	0.0756	0.9395
0	1	0.0722	0.9244	-0.0211	1.0215	0.0012	1.0278	0.0561	0.9389
0	1	0.0724	0.9272	-0.0214	1.0195	-0.0764	1.0618	0.0761	0.9372
0	1	0.0775	0.9182	0.0412	0.9542	-0.0762	1.0598	0.0752	0.9390
0	1	0.0719	0.9277	-0.0732	1.0719	0.1756	0.8180	0.0758	0.9389
0	1	0.0703	0.9334	-0.0804	1.0794	0.2449	0.7672	0.0573	0.9383
0	1	0.0708	0.9342	-0.1015	1.0986	0.2294	0.8003	0.0437	0.9390
0	1	0.0775	0.9171	0.0080	0.9911	-0.1165	1.0942	0.0763	0.9404
0	1	0.0359	0.9600	0.0099	0.9907	0.0301	0.9714	0.0747	0.9398
0	1	0.0858	0.9132	0.0120	0.9884	-0.0708	1.0879	0.0733	0.9390
0	1	0.0753	0.9286	-0.0128	1.0140	-0.0401	1.0378	0.0764	0.9398
0	1	0.0585	0.9395	0.0136	0.9895	0.0283	0.9805	0.0760	0.9372
0	1	0.0709	0.9292	-0.0601	1.0585	0.0515	0.9567	0.0764	0.9386